



University of Nicosia, Cyprus

Course Code COMP-112	Course Title Software Development Lab I	ECTS Credits 4
Department Computer Science	Semester Fall, Spring	Prerequisites None
Type of Course Required	Field Computer Science	Language of Instruction English
Level of Course 1 st Cycle	Year of Study 1 st	Lecturer(s) Dr Andreas Savva
Mode of Delivery Face-to-face	Work Placement N/A	Co-requisites COMP-111

Objectives of the Course:

The main objectives of the course are to:

- introduce to the students good software development practices
- provide practical experience in developing software with appropriate comments and comment tags
- provide practical experience in developing readable, maintainable, robust, and secure source code
- provide practical experience in developing software which checks all function arguments and the function return argument
- provide practical experience in developing function/method tests and automatic test suites
- introduce tools/environments which automatically can run test suites
- introduce tools which automatically check the quality of the code
- introduce environments which provide code check-style

Learning Outcomes:

After completion of the course students are expected to be able to:

1. Be proficient in developing high quality source code.
2. Describe what high quality source code is.
3. Demonstrate the ability to use tools to run automatic test suites.
4. Demonstrate the ability to use tools in order to test the quality and/or complexity of source code.
5. Be proficient in using development environments, which provides check-styles, and other tools for developing high quality source code.

Course Contents:

1. Introduce the concept of developing high quality source code.
2. Develop source code with appropriate comments, comment blocks, and comment tags.
3. How to develop readable, maintainable, robust, and secure source code.
4. How to protect functions, methods, and the application by checking the state of incoming and/or outgoing arguments.
5. Use features and/or programming patterns of the programming language (like

const, assert, safe-casting mechanisms) in order to increase quality of the source code.

6. Develop test cases and test suites to verify the correctness of the source code and apply automatic testing tools.
7. Use tools in order to check code complexity, dead-code, and other unwanted elements of the developed source code.
8. Introduction to code style and how to set up automatic check-style tools within the development environment to enforce/police the coding-style.

Learning Activities and Teaching Methods:

Lectures, lab presentations, lab tutorials, practical exercises, assignments

Assessment Methods:

Homework, projects

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Gary J. Bronson	Program Development and Design using C++	Thomson Course Technology	2006	0-619-21677-8

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Matthew B. Doar	Practical Development Environments	O'Reilly Media	2005	0-596-007965